Lie group techniques for Neural Learning

Edinburgh June 2004

Elena Celledoni

SINTEF Applied Mathematics, IMF-NTNU

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
03 JAN 2005	**N/A**	**-**

4. TITLE AND SUBTITLE
Lie group techniques for Neural Learning

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SINTEF Applied Mathematics, IMF-NTNU

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001749, Lie Group Methods And Control Theory Workshop Held on 28 June 2004 - 1 July 2004., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**36**	

Outline

- **Neural Networks**
  - a short introduction

Outline

- **Neural Networks**
  - a short introduction
- **Independent Component Analysis**
  - Stochastic signal processing
  - Constraint optimization in ICA

Outline

- **Neural Networks**
  - a short introduction
- **Independent Component Analysis**
  - Stochastic signal processing
  - Constraint optimization in ICA
- **Geometric Integration of Learning equations**
  - gradient flows and algorithms on manifolds
  - MEC learning
  - Newton methods
  - diffusion algorithms

Neural Networks

Goals:

- Achieve efficient use of machines in tasks currently solved by humans
- Improve computing capabilities looking at the brain as a model
- Understand how the brain works

Applications

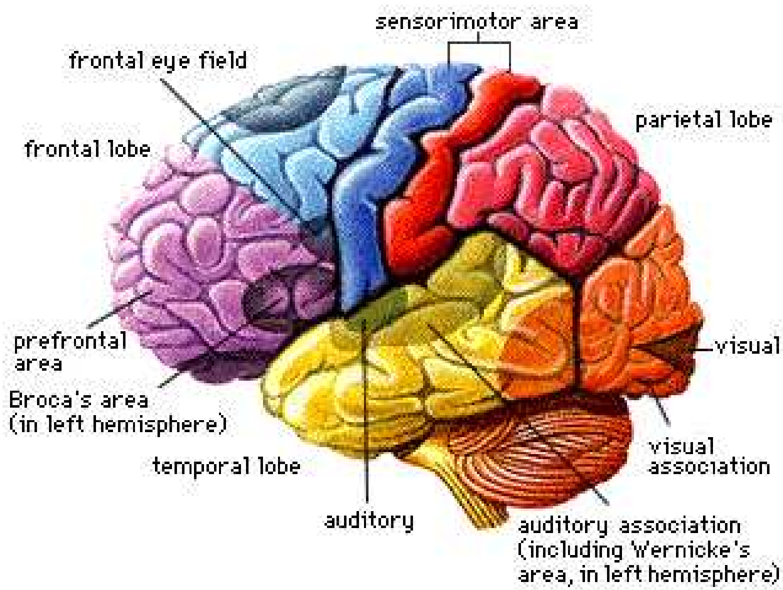
- Machine Learning
  1. How can a computer learn from a set of examples?
  2. Constraint optimization
  3. Pattern recognition, classification
  4. Associative memory
- Cognitive science
  1. Models for high level reasoning: language, problem solving
  2. Models for low level reasoning: vision, speech recognition, speech generation
- Neurobiology: find models for how the brain works

List of fields where Neural Networks are used

- Signal processing
- Control
- Robotics (navigation, vision)
- Medicine
- Business and Finance
- Data Compression

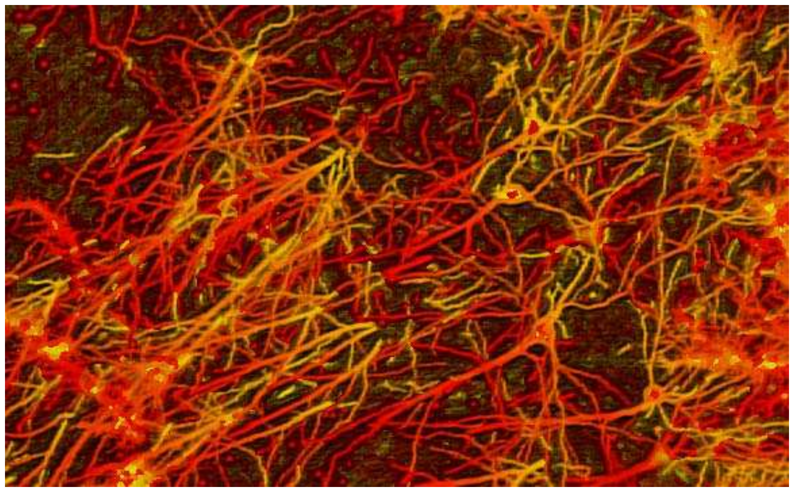
The brain as an Information Processing System

- Massively parallel: 10 billion neurons, 10000 synapses per neuron
- Slow hardware: neurons operate at about 100 Hz, while conventional CPUs execute several hundred million machine level operations per second



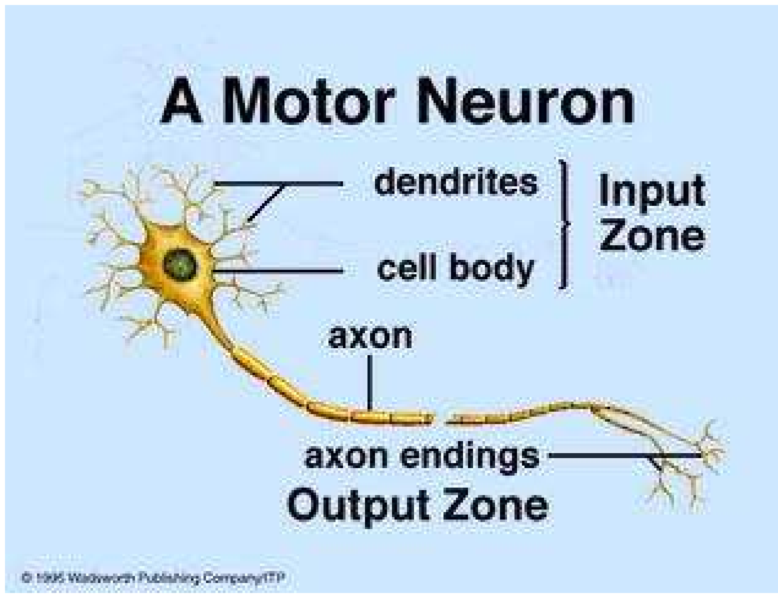
The brain as an Information Processing System

- Massively parallel: 10 billion neurons, 10000 synapses per neuron
- Slow hardware: neurons operate at about 100 Hz, while conventional CPUs execute several hundred million machine level operations per second



The brain as an Information Processing System

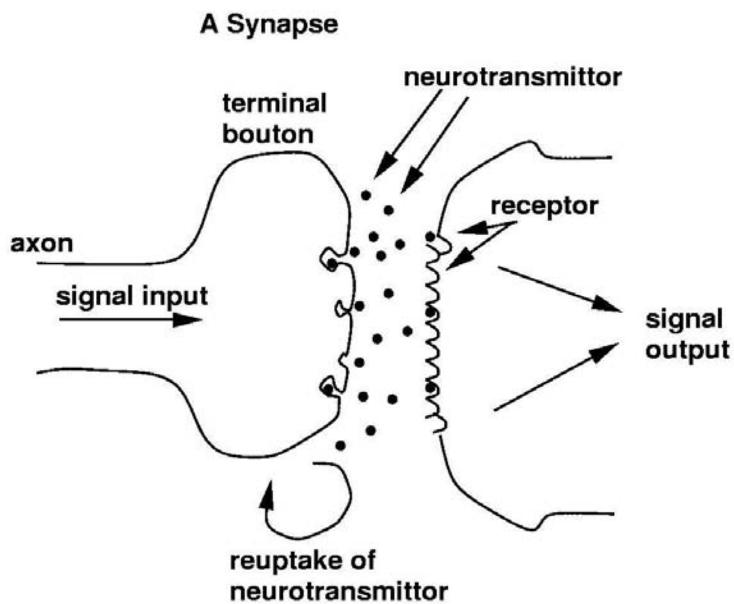
- Massively parallel: 10 billion neurons, 10000 synapses per neuron
- Slow hardware: neurons operate at about 100 Hz, while conventional CPUs execute several hundred million machine level operations per second



The brain as an Information Processing System

- Massively parallel: 10 billion neurons, 10000 synapses per neuron
- Slow hardware: neurons operate at about 100 Hz, while conventional CPUs execute several hundred million machine level operations per second

Synapse: transmission of a signal between neurons via a neurotransmitter. **Learning** corresponds to alteration of the strength of the connection between neurons.

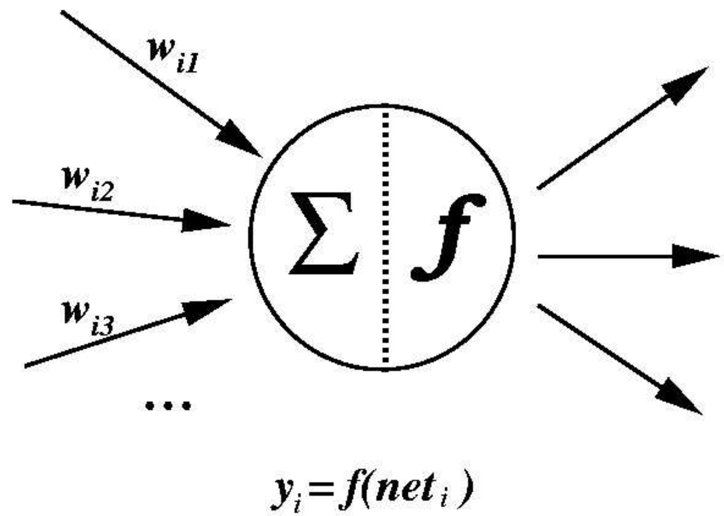


A simple model for a neuron

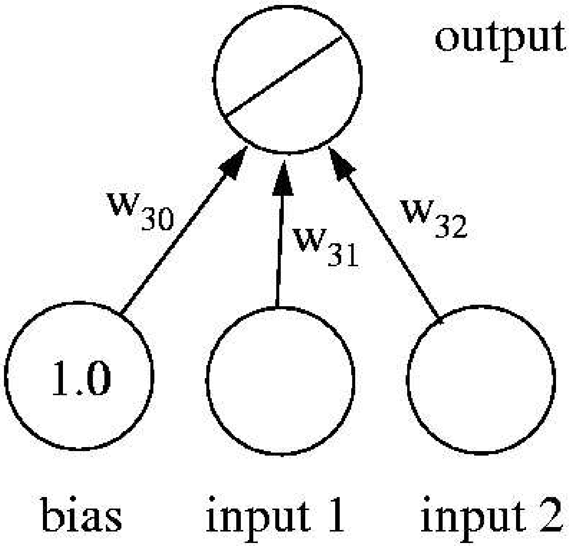
Each node (neuron) receives signal inputs form n neighbor nodes.

$$y_i = f(\sum_j w_{i,j} y_j)$$

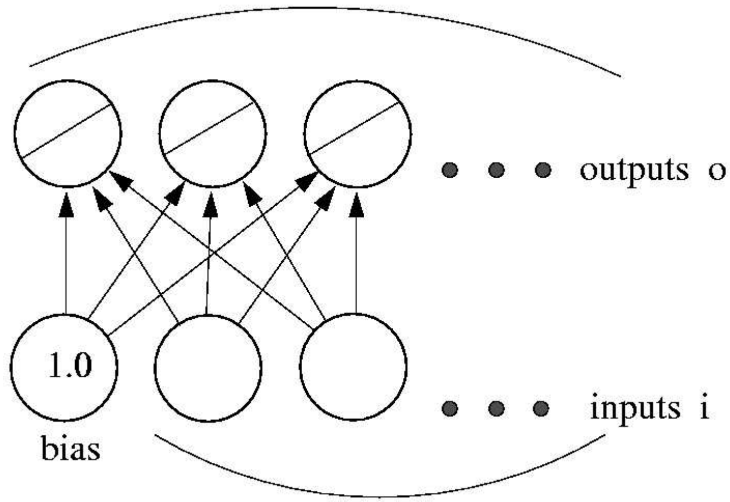
The weighted sum $\sum_j w_{i,j} y_j$ is called the net input. f is the activation function, if f is the identity we have a **linear unit**. y_i is the output signal



Linear Neural Networks



Several inputs one output



n inputs p outputs

http://www.willamette.edu/ gorr

Independent Component Analysis

The cocktail-party problem

Suppose you record two time signals $x_1(t)$ and $x_2(t)$ form two different positions in a room. Each recorded signal is a linear mixture of the voices of two speakers which emit two sources $s_1(t)$ and $s_2(t)$

$$\begin{aligned} x_1(t) &= a_{1,1}s_1(t) + a_{1,2}s_2(t) \\ x_2(t) &= a_{2,1}s_1(t) + a_{2,2}s_2(t) \end{aligned}$$

Estimate $s_1(t)$ and $s_2(t)$ from the sole knowledge of $x_1(t)$ and $x_2(t)$

Independent Component Analysis

The cocktail-party problem

Suppose you record two time signals $x_1(t)$ and $x_2(t)$ form two different positions in a room. Each recorded signal is a linear mixture of the voices of two speakers which emit two sources $s_1(t)$ and $s_2(t)$

$$\begin{aligned} x_1(t) &= a_{1,1}s_1(t) + a_{1,2}s_2(t) \\ x_2(t) &= a_{2,1}s_1(t) + a_{2,2}s_2(t) \end{aligned}$$

Estimate $s_1(t)$ and $s_2(t)$ from the sole knowledge of $x_1(t)$ and $x_2(t)$

Assume the sources and the recorded signals are samples of the zero-mean random variables x_1, x_2, (**mixtures**) and s_1, s_2 (**independent components**).

Assumption $s_1(t)$ and $s_2(t)$ are statistically independent

Independent Component Analysis

The cocktail-party problem
Suppose you record two time signals $x_1(t)$ and $x_2(t)$ form two different positions in a room. Each recorded signal is a linear mixture of the voices of two speakers which emit two sources $s_1(t)$ and $s_2(t)$

$$\begin{aligned} x_1(t) &= a_{1,1}s_1(t) + a_{1,2}s_2(t) \\ x_2(t) &= a_{2,1}s_1(t) + a_{2,2}s_2(t) \end{aligned}$$

Estimate $s_1(t)$ and $s_2(t)$ from the sole knowledge of $x_1(t)$ and $x_2(t)$
Assume the sources and the recorded signals are samples of the zero-mean random variables x_1, x_2, (**mixtures**) and s_1, s_2 (**independent components**).

Assumption $s_1(t)$ and $s_2(t)$ are statistically independent

Unknown source signals $\mathbf{s}(t) = [s_1(t), \ldots, s_n(t)]^{\mathrm{T}}$
Given the output signals $\mathbf{x}(t) = A\mathbf{s}(t)$, $\mathbf{x}(t) = [x_1(t), \ldots, x_k(t)]^{\mathrm{T}}$
Unknown mixing matrix A $p \times n$

Independent Component Analysis

The cocktail-party problem

Suppose you record two time signals $x_1(t)$ and $x_2(t)$ form two different positions in a room. Each recorded signal is a linear mixture of the voices of two speakers which emit two sources $s_1(t)$ and $s_2(t)$

$$\begin{aligned} x_1(t) &= a_{1,1}s_1(t) + a_{1,2}s_2(t) \\ x_2(t) &= a_{2,1}s_1(t) + a_{2,2}s_2(t) \end{aligned}$$

Estimate $s_1(t)$ and $s_2(t)$ from the sole knowledge of $x_1(t)$ and $x_2(t)$

Assume the sources and the recorded signals are samples of the zero-mean random variables x_1, x_2, **(mixtures)** and s_1, s_2 **(independent components)**.

Assumption $s_1(t)$ and $s_2(t)$ are statistically independent

Unknown source signals $\mathbf{s}(t) = [s_1(t), \ldots, s_n(t)]^{\mathrm{T}}$

Given the output signals $\mathbf{x}(t) = A\mathbf{s}(t)$, $\mathbf{x}(t) = [x_1(t), \ldots, x_k(t)]^{\mathrm{T}}$

Unknown mixing matrix A $p \times n$

Find approximations $\mathbf{y}$ of $\mathbf{s}$ by constructing a de-mixing matrix W and

$$\mathbf{y} = W\mathbf{x}.$$

Principles for reconstruction

The sum of two independent random variables usually has distribution closer to Gaussian than the two original random variables. (**Central Limit Theorem**)

$$\mathbf{x} = A\mathbf{s}$$

Find

$$\mathbf{y} = W\mathbf{x} \approx \mathbf{s}$$

maximizing nongaussianity.

A measure of nongaussianity is kurtosis,

$$\mathrm{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2,$$

with y of unit variance $\mathrm{kurt}(y) = E\{y^4\} - 3$.

Withening

Preprocessing of the output signals $\mathbf{x} \to \tilde{\mathbf{x}}$ such that the components of $\tilde{\mathbf{x}}$ are uncorrelated with variances equal to 1

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I}.$$

Withening

Preprocessing of the output signals $\mathbf{x} \to \tilde{\mathbf{x}}$ such that the components of $\tilde{\mathbf{x}}$ are uncorrelated with variances equal to 1

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I}.$$

Use for example $E\{\mathbf{x}\mathbf{x}^T\} = VDV^T$ and

$$\tilde{\mathbf{x}} = VD^{-1/2}V^T\mathbf{x} \;\Rightarrow\; E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I}$$

and $\tilde{\mathbf{x}} = VD^{-1/2}V^T A\mathbf{s} = \tilde{A}\mathbf{s}$, then

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \tilde{A}E\{\mathbf{s}\mathbf{s}^T\}\tilde{A}^T = \tilde{A}\tilde{A}^T = \mathbf{I}$$

Reconstruction

Reconstruction of $\mathbf{s}$. We can look for a de-mixing matrix W s.t. $W^T W = I_p$ and $y(t) = W x(t)$ solving

$$\min_{W^T W = I_p} D(W)$$

$D(W)$ is the dependency among the components.

A. Hyvärinen and E. Oja Independent component analysis: A tutorial, *Neural Networks*.

Optimizing via gradient flows

Let $\mathcal{M}$ be a Reimannian manifold with metric $m(\cdot,\cdot)$, given $\phi : \mathcal{M} \to \mathbb{R}$ a smooth function the equilibria of

$$\dot{x}(t) = -\mathrm{grad}\phi\big(x(t)\big)$$

are the critical points of ϕ.
$\mathrm{grad}\phi$ is such that:

- $\mathrm{grad}\phi(x) \in T_x\mathcal{M}$
- $\phi'|_x\,(v) = m(\mathrm{grad}\phi(x), v)$ for all $v \in T_x\mathcal{M}$

Optimizing via gradient flows

Let $\mathcal{M}$ be a Reimannian manifold with metric $m(\cdot,\cdot)$, given $\phi : \mathcal{M} \to \mathbb{R}$ a smooth function the equilibria of

$$\dot{x}(t) = -\mathrm{grad}\phi\big(x(t)\big)$$

are the critical points of ϕ.
$\mathrm{grad}\phi$ is such that:

- $\mathrm{grad}\phi(x) \in T_x\mathcal{M}$
- $\phi'|_x(v) = m(\mathrm{grad}\phi(x), v)$ for all $v \in T_x\mathcal{M}$

U. Helmke and J.B. Moore, Optimization and Dynamical Systems, Springer-Verlag 1994

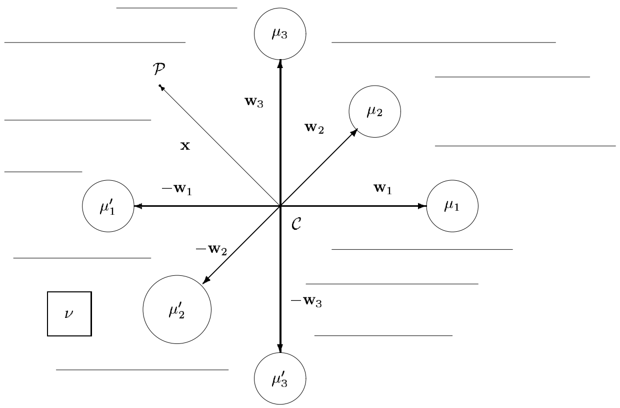
M.T. Chu and K.R. Drissel, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM J. Num. Anal., 1990

S.I. Amari, *Natural Gradient Works Efficiently in Learning*, Neural Computation, 1998

Y. Nishimori, *Learning algorithm for ICA by geodesic flows on orthogonal* Proc. IJCNN 99

Optimizing via mechanical systems I

Consider $\mathcal{S}^* = \{[2m_i, \mathbf{w}_i]\}$ rigid system of n masses m_i with positions $\mathbf{w}_i$ (unitary distance form the origin on mutually orthogonal axis). The masses move in a viscous liquid. No translation.



$$
\begin{aligned}
\dot{W} &= HW, \quad P = -\mu HW \\
\dot{H} &= \frac{1}{4}\left[\left[F+P\right)W^T - W(F+P)^T\right]
\end{aligned}
$$

μ viscosity parameter
W matrix of the positions
H angular velocity matrix
P matrix of the viscosity resistance
F active forces

W is on $O(n)$ or on the Stiefel manifold

Optimizing via mechanical systems II

The mechanical system seen as an adapting rule for neural layers with weight matrix W.
The forces

$$F := -\frac{\partial U}{\partial W}$$

with U a **potential energy function**. The equilibria of the mechanical systems $\mathcal{S}^*$ are at the local minima of U.
Take $U = J_C$ cost function to be minimized, or $U = -J_O$ objective function to be maximized, $W(t), \;\; t \to \infty$ approaches the solution of the optimization problem.

S. Fiori, *'Mechanical' Neural Learning for Blind Source Separation*, Electronics Letters, 1999

Reformulation of the equations when $n << p$

Using the Lie algebra

$$\begin{aligned} \dot{W} &= HW, \quad P = -\mu HW \\ \dot{H} &= \tfrac{1}{4}\big[\big[F+P\big)W^T - W(F+P)^T\big] \end{aligned}$$

Using the tangent space

$$\begin{aligned} \dot{W} &= V \\ \dot{V} &= g(V,W) \end{aligned}$$

where

$$V = (GW^T - WG^T)W, \qquad G = V - W(W^T V/2 + S)$$

and

$$g(V,W) = (LW^T - WL^T)W + (GW^T - WG^T)V, \qquad L = \dot{G} - GW^T G$$

The learning algorithm

$$\begin{cases} V_{n+1} &= V_n + hg(V_n, W_n) \\ G_n &= V_n - 1/2 W_n (W_n^T V_n) \\ W_{n+1} &= \exp(h(G_n W_n^T - W_n G_n^T)) W_n \end{cases}$$

with $W_0 = I_{n \times p}$ and $V_0 = 0_{n \times p}$.
Here

$$\exp(h(G_n W_n^T - W_n G_n^T)) = [W_n, W_n^{\perp}] \exp\left(\begin{bmatrix} C - C^T & -R^T \\ R & O \end{bmatrix}\right) [W_n, W_n^{\perp}]^T$$

and $C = W_n^T G_n$, and $G_n - W_n C = W_n^{\perp} R$. We exponentiate matrices of dimension $2p \times 2p$ instead of $n \times n$.

Computational cost

For the exponential $9np^2 + np + \mathcal{O}(p^3)$ flops. For the overall geodesic learning algorithm (one step) $21np^2 + 6np + \mathcal{O}(p^3)$ flops.

Computational gain

Computing the largest eigenvalue of an $n \times n$ matrix A (discretization of the 1-D Laplacian with finite differences).

The potential energy function is $U(w) = -w^T A w$, $p = 1$.

SIZE OF A	New MEC	Old MEC
$n = 32$	4.72×10^5	1.31×10^6
$n = 64$	1.82×10^6	5.25×10^6
$n = 128$	7.39×10^6	2.10×10^7
$n = 256$	2.49×10^7	8.39×10^7

Floating point operations per iteration versus the size of the problem.

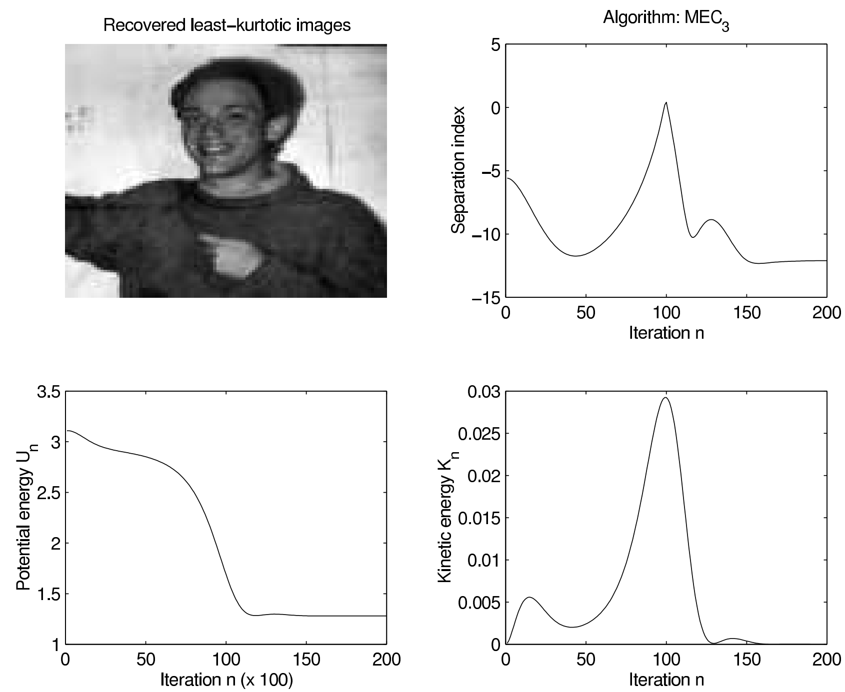
Experiments Blind source separation

Original images, with their kurtosis and their linear mixtures

Source separation

The force $F(W) = -kE_x[x(x^T W)^3_{.}]$.

Recovered image, and potential energy



References

- E. Celledoni and S. Fiori, *Neural learning by Geometric Integration of Reduced 'Rigid-Body' Equations*, J. CAM to appear.
- E. Celledoni and B. Owren, *On the implementation of Lie group methods on the Stiefel manifold*, Numerical Algorithms, 2003.

Future work

- On the orthogonal group consider quasi-geodesic paths using low-rank splittings
- Other manifolds occur in the case of multi-layer neural networks: Flag manifolds
- comparison with Newton methods

Newton methods, Mahony's approach

For finding minima or maxima of $\phi : \mathcal{G} \to \mathbb{R}$, and $\mathcal{G}$ is a Lie group,

- choose an inner product $< \cdot, \cdot >$ on the Lie algebra $\mathfrak{g}$ and take an orthonormal basis in the Lie algebra $X_1, \ldots, X_d$, and $\tilde{X}_1, \ldots, \tilde{X}_d$ the right invariant vector fields
- $$\mathrm{grad}\phi = \sum_{i=1}^{d} m(\tilde{X}_i, \mathrm{grad}\phi)\tilde{X}_i = \sum_{i=1}^{d} (\tilde{X}_i\phi)\tilde{X}_i$$

  ($m(\tilde{X}, \tilde{Y}) =< X, Y >$ (right invariant group metric))
- if $\exp(X)\sigma$ is a critical point of ϕ, the vector field $\tilde{X}$ satisfies,

$$\mathrm{grad}\phi(\sigma) + \mathrm{grad}(\tilde{X}\phi)(\sigma) = 0$$

R. E. Mahony *The constrained Newton method on a Lie group and the symmetric eigenvalue problem* , Lin. Alg. and Appl. 1996

Newton methods, Mahony's approach

For finding minima or maxima of $\phi : \mathcal{G} \to \mathbb{R}$, and $\mathcal{G}$ is a Lie group,

- choose an inner product $< \cdot, \cdot >$ on the Lie algebra $\mathfrak{g}$ and take an orthonormal basis in the Lie algebra $X_1, \ldots, X_d$, and $\tilde{X}_1, \ldots, \tilde{X}_d$ the right invariant vector fields
- $$\mathrm{grad}\phi = \sum_{i=1}^{d} m(\tilde{X}_i, \mathrm{grad}\phi)\tilde{X}_i = \sum_{i=1}^{d} (\tilde{X}_i \phi)\tilde{X}_i$$

  ($m(\tilde{X}, \tilde{Y}) =< X, Y >$ (right invariant group metric))
- Find X^k such that $\tilde{X}^k$ solves

  $$\mathrm{grad}\phi(\sigma_k) + \mathrm{grad}(\tilde{X}^k \phi)(\sigma_k) = 0$$

  set $\sigma_{k+1} = \exp(X^k)\sigma_k$, $k \leftarrow k+1$ and continue, (equivalent to Lie Euler for $\dot{\sigma} = X^k \sigma$, $\sigma(0) = \sigma^k$)

R. E. Mahony *The constrained Newton method on a Lie group and the symmetric eigenvalue problem* , Lin. Alg. and Appl. 1996

Newton methods, other approaches

- A. Edelman, T. Arias, S.T. Smith, *The geometry of Algorithms with orthogonality constrains*, SIAM J. Matrix Anal. Newton methods and Conjugate Gradient on the Stiefel and Grassman manifolds.
- B. Owren and B. Welfert, *The Newton iteration on Lie groups*, BIT 2000. Context: implicit Lie group methods, this method can be applied directly in the implicit integration of gradient flows
- L. Lopez, C. Mastroserio, T. Politi. *Newton-type methods for solving nonlinear equations on quadratic matrix groups.* J. CAM 2000. Similar as previous one, using the Cayley transformation
- J.P. Dedieu and D. Nowicki, *Symplectic methods for the approximation of the exponential and the Newton sequence on Reimannian submanifolds*, Preprint february 2004. General Reimanninan manifold, use of tangent space parametrizations, geodesic seen as the trajectory of a free particle attached to the manifold

Diffusion-type algorithms

Perturbation of the standard Reimannian gradient to obtain a rondomized gradient. Diffusion-type gradient on $\mathfrak{SO}(n)$

$$V_{\mathrm{diff}}(t) = V(t) + \sqrt{2\theta} \sum_{k=1}^{n(n-1)/2} X_k \frac{d\mathcal{W}_k}{dt}$$

$V(t)$ deterministic gradient, X_k is a basis of the Lie algebra $\mathfrak{so}(n)$ orthogonal with respect to the chosen metric, and $\mathcal{W}_k(t)$ are real-valued, independent standard Wiener processes i.e. a random variable $\mathcal{W}$ continuous in t s.t.

- $\mathcal{W}(0) = 0$ with probability 1
- for $0 \leq \tau < t$ the random variable $\mathcal{W}(t) - \mathcal{W}(\tau)$ is normally distributed with mean zero and variance $t - \tau$
- for $0 \leq \tau < t < u < v$, the increments $\mathcal{W}(t) - \mathcal{W}(\tau)$ and $\mathcal{W}(v) - \mathcal{W}(u)$ are statistically independent

Diffusion-type algorithms

Perturbation of the standard Reimannian gradient to obtain a rondomized gradient. Diffusion-type gradient on $\mathfrak{SO}(n)$

$$V_{\text{diff}}(t) = V(t) + \sqrt{2\theta} \sum_{k=1}^{n(n-1)/2} X_k \frac{d\mathcal{W}_k}{dt}$$

$V(t)$ deterministic gradient, X_k is a basis of the Lie algebra $\mathfrak{SO}(n)$ orthogonal with respect to the chosen metric, and $\mathcal{W}_k(t)$ are real-valued, independent standard Wiener processes The learning differential equation is

$$\frac{dW}{dt} = -V_{\text{diff}}(t))W$$

Langevin-type stochastic differential equation on the orthogonal group

X. Liu, A. Srivastava, K. Galivan, *Optimal linear representation of images for object recognition*, IEEE Trans. Pattern Analysis, 2004.

Conclusion

- Integration of learning equations and gradient flows is achieved with simple first order explicit Lie group integrators
- Efficient approximation of the matrix exponential from a Lie algebra to a Lie group or the computation of geodesics is crucial
- Development of methods based on other coordinate maps then the exponential, and quasi-geodesic strategies
- Geometric integration of stochastic differential equations